FACIAL EMOTION RECOGNITION

Amjed Abbas Ahmed, Ali Abbas AL-Shukrawi, Ahmed Mahdi Aboud, Ahmad Hassan Laibi

Faculty of Information Science and Technology, University Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Imam Al kadhum college(IKC),Iraq –Baghdad

ABSTRACT:

Researchers that is discipline are interested in creating methods to decipher, encode, and extract these characteristics from facial expressions for better computer prediction. Deep learning has been incredibly successful, and several deep learning architectures are being used to improve performance. This project's goal is to do research on current advances in automatic facial emotion recognition (FER) using deep learning, which has applications in many areas, such as human-machine interactions, health, and safety. We provide a deep learning method that may concentrate on the key features of the face and uses a convolutional neural network as its foundation. The experiment used the CK +48 database, and the average accuracy for problems with emotion detection was 98.6%.

Keywords: *Facial Emotion Recognition, Algorithm, Deep learning , CK.*

INTRODUCTION

Communication between people always involves emotions. They may or may not be visible to the unaided eye and can manifest in a wide variety of ways. Therefore, with the appropriate equipment, any signs that precede or follow them are susceptible to detection and identification. There are more now than ever before, and there is greater awareness of the importance of understanding a person's feelings. A multitude of fields, such as but not limited to human-computer interfaces, animation, security, and autism diagnosis, may recognize human emotions. Urban sound perception and children with Autism Spectrum Disorders (ASD). Different indicators, including Emotions can be recognized through the use of text, audio, EEG, and even facial expressions. People frequently notice facial expressions the most, if not the most, of these qualities for a variety of reasons.: Compared to other methods for human recognition, faces are more accessible, have a variety of traits that are relevant for emotion recognition, and are simpler to gather big datasets of [1]. Two independent but related research fields in automatic emotion identification are artificial intelligence (AI) and psychology human emotion perception. Verbal and nonverbal information gathered by a variety of sensors, such as changes in facial expression, voice tonality, and physiological indications, can be used to infer human emotional states. [2].

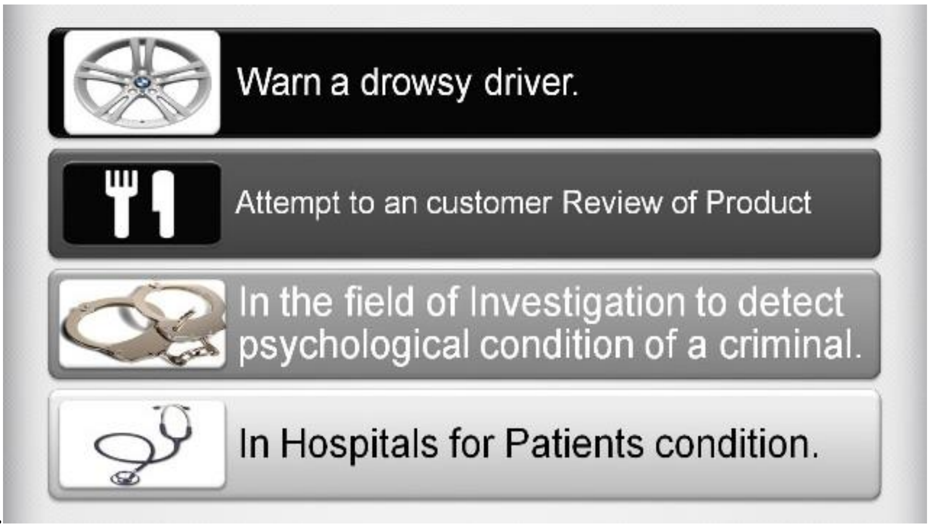

Figure 1 Emotion recognition applications.

LITERATURE REVIEW

Some of the most recent convolutional neural network-based expression recognition researches are discussed.

The multi-region ensemble convolutional neural network approach for face expression identification was developed by Yingruo Fan et al. [3]. The features derived from the three sites of the eyes, nose, and mouth are sent to three sub-networks. Then, in order to forecast emotions, the weights from three subnetworks are combined. This investigation used the databases RAF-DB and AFEW 7.0.

Yingying Wang et al. proposed the auxiliary model as a technique for identifying emotions. In order to collect as much data as feasible for this study, the weighting method is utilized to combine the data from the eyes, nose, and mouth with the overall facial image [4]. The model is examined using the CK, FER2013, SFEW, and JAFFE databases. VGG16 and Resnet50 were used by Frans Norden et al. to demonstrate face expression recognition [5]. The databases JAFFE and FER2013 were used in this analysis. For identifying emotions, Nithya Roopa and associates proposed the Inception V3 model [7]. The work was evaluated with a test accuracy of 39% using the KDEF database. Sreelakshmi et al. [8] created an emotion identification system based on the MobileNet V2 architecture to manage occlusions and location fluctuations.

When evaluated on actual occluded photos, the model achieves a 92.5 percent accuracy rate. CNN's pre-trained features based on facial expression detection were given by Aravind Ravi. [9]. To predict the expressions, a support vector machine is employed. once the features have been obtained using a pre-trained VGG19 network. The accuracy rates of JAFFE and CK, the two databases utilized in the experiment, were 92.86 and 92.26 percent, respectively. A support vector machine classifier-based transfer learning technique was presented by Shamoil Shaees et al. [10]. CNNs and AlexNet are used in this study to recover features, which are subsequently fed into an SVM for classification. The study made use of the CK? and NVIE databases, and the results were quite accurate. Convolution neural networks were used by the authors of [11] to propose a solution for face emotion identification. The experiment made use of the fer2013 dataset and a number of models, including VGG 19, VGG 16, and ResNet50. With a score of 63.07 percent, the VGG 16 model has the best accuracy rate of the three. The LeNet-based system for emotion recognition was developed by Mehmet Akif OZDEMIR et al.

AIM OF THE STUDY

1. Our objective in this system is to recognize the emotion of the person automatically in a real photo.

2 . The capacity to replicate human coding skills Nonverbal communication cues are communicated through gestures and facial expressions, which are crucial in interpersonal interactions.

3. Demonstrating CNN's ability to accurately categorize facial emotions.

EMOTION’S THEORIES

The three categories of emotional explanations are depicted in Figure 2: physiologic James-Lange and Cannon-Bard theories, cognitive Lazarus theories, and neurological explanations (Facial feedback theory).According to the James-Lange paradigm, emotion is a result of how one interprets a bodily response. Following this, Walter Cannon disproved the James- Lange theory and created the Cannon- Bard hypothesis [16], which holds that emotions and physiological reactions occur simultaneously. The Lazarus theory, also known as the cognitive appraisal theory, states that a person's consideration of their body's physiological response to experiencing the emotion comes next [17]. Last but not least, the facial feedback theory explains how facial expressions can convey emotional experience.

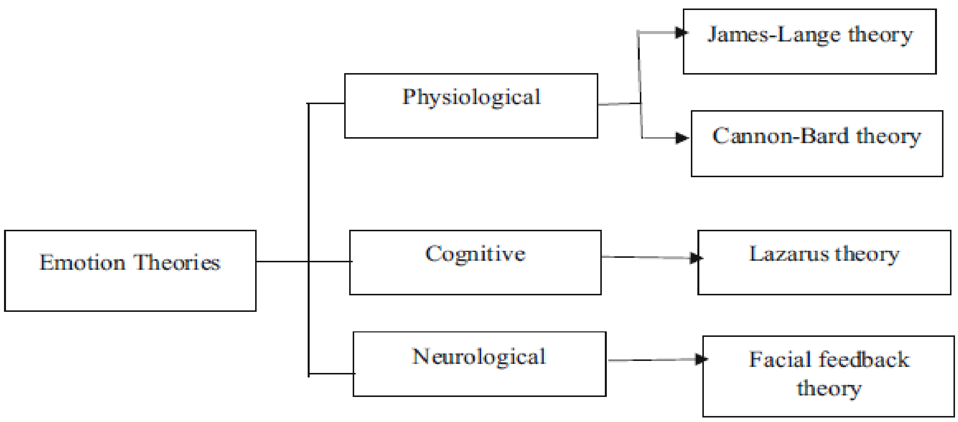

Figure 2 Emotions theories.

(IJAER) 2022, Vol. No. 24, Issue No. II, August **e-ISSN: 2231-5152, p-ISSN: 2454-1796**

EMOTIONS MODEL

The two most typical forms of emotion models are categorical and dimensional. According to Ekman and Friesen [18], the category model represents the fundamental emotions of disgust, fear, sadness, pleasure, and anger. Emotions are represented utilizing a two- or three-dimensional model (Arousal and Valence) (Power, Arousal, and Valence). In Figure 3, the emotional model is shown. The valence of an emotion determines whether it is positive or negative, and its arousal measures how excited an expression is. Models in two dimensions such the PANA, vector, and circumplex are available (Positive Action- Negative Action).

Plutchik devised the three-dimensional models known as Pleasure, Arousal, and Dominance (PAD) [19].

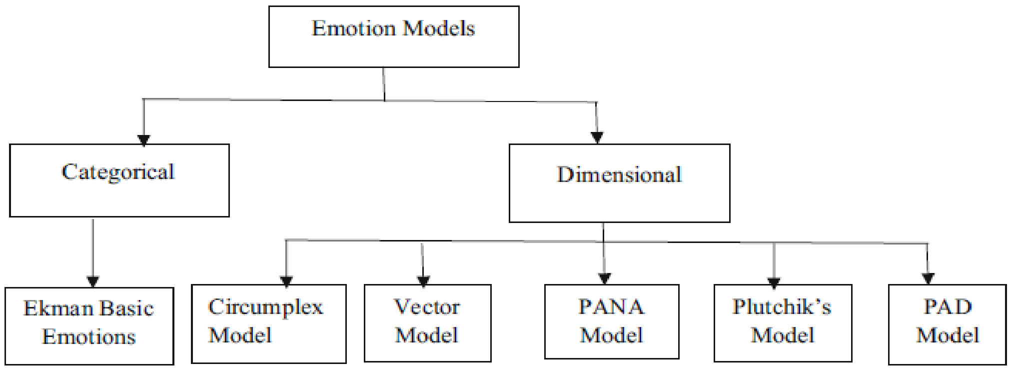

Figure 3 Emotions models.

DEEP LEARNING TECHNIQUES

A machine learning approach called deep learning teaches computers to perform activities that people do automatically. The ability of driverless cars to tell a pedestrian from a lamppost and recognize stop signs depends heavily on deep learning. Consumer electronics like smartphones, tablets, TVs, and hands-free speakers all require it for voice control. Recent years have seen a lot of interest in deep learning, and for good reason. Results that were previously impossible are now being achieved [20].

HOW DEEP LEARNING WORKS

Because most deep learning methods rely on neural network topologies, deep learning models are sometimes referred to as deep neural networks.

The term "deep" usually denotes the number of hidden layers in the neural network. Unlike regular neural networks, which typically only have two or three layers, deep neural networks can contain as many as 150 hidden layers. Deep learning models are trained using enormous amounts of labeled data and neural network topologies that automatically extract features from the data. Figure 4 [2] illustrates a network with three hidden levels.

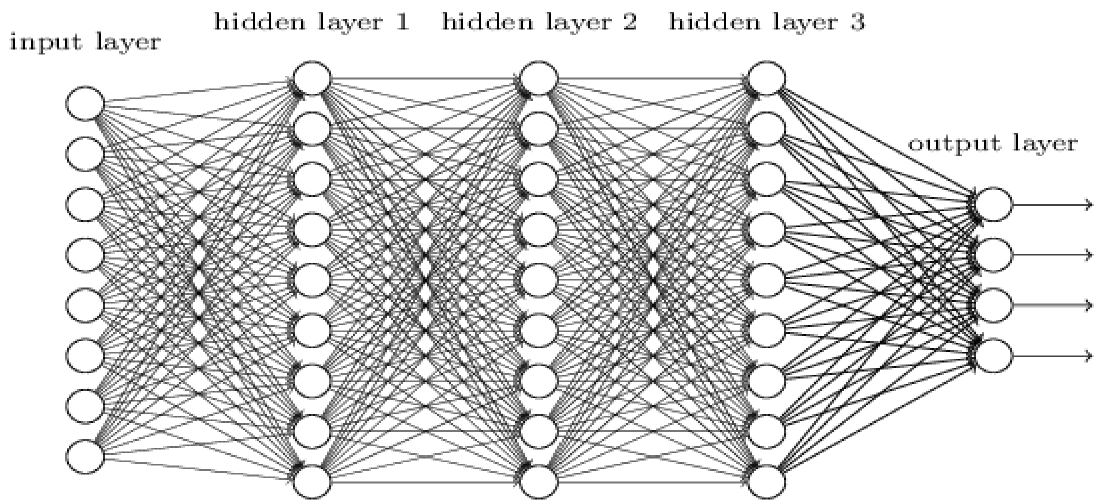

Figure 4 Neural network with three hidden layers.

CNN is one of the most popular types of deep neural networks (CNN or ConvNet). In order to handle 2D data, including images, a CNN employs 2D convolutional layers and blends learned features with input data [21]. Because CNNs perform the manual feature extraction for you, you do not need to be familiar with the features that are used to categorize images. CNN uses direct feature extraction from images. Instead of being pre-trained, the relevant features are found while the network is trained on a set of photos. This automatic feature extraction makes deep learning models incredibly accurate for computer vision applications like object categorization. The graphic below depicts a typical CNN process [21].

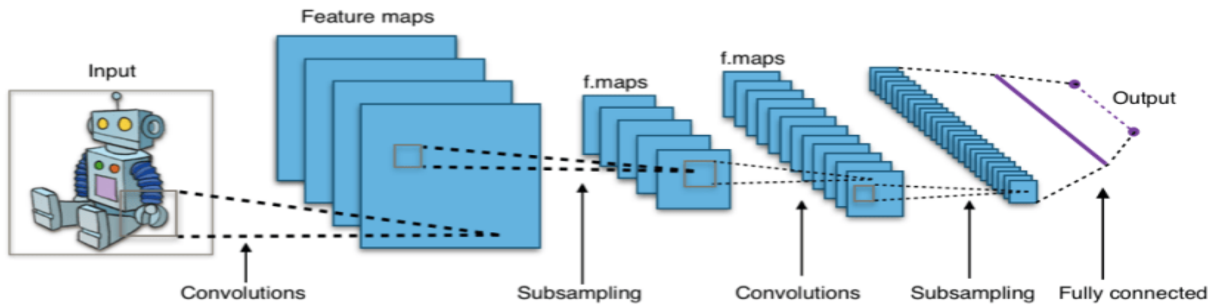

Figure 5 Standard CNN process.

The convolutional layer, which is seen in figure 5 above, is the fundamental component of a CNN. A group of learnable filters (also known as kernels) that have a narrow receptive field yet cover the entire depth of the input volume make up the layer's parameters. Each filter is convolved across the width and height of the input volume during the forward pass, and the dot product between each filter entry and the input is computed to create a 2-dimensional activation map of the filter. As a result, the network recognizes filters that turn on when it recognizes a certain feature at a particular location in the input. In CNN, a convolution layer is frequently followed by a pooling or subsampling layer.

PROPOSED WORK INTRODUCTION

Figure 6 represents the block diagram of the method.

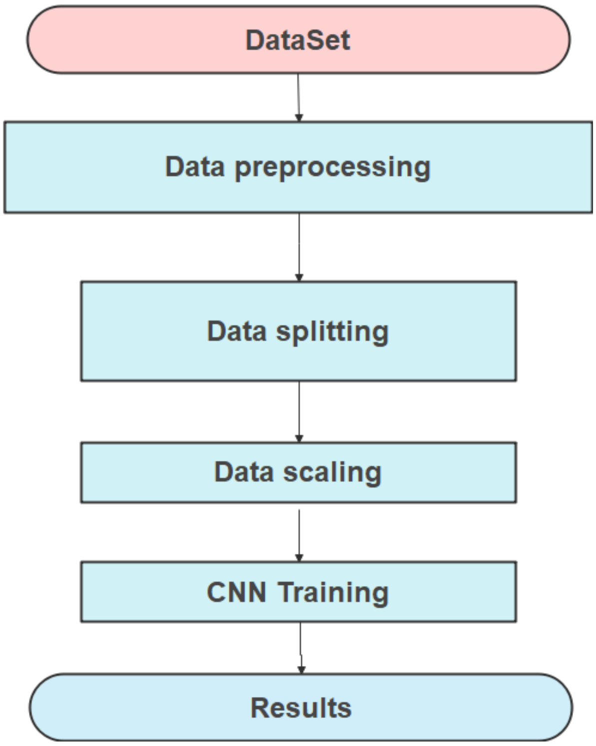

Figure 6 Flowchart of the proposed techniques

DATASET DESCRIPTION

This is a cropped version of ck+ 48 dataset with only 5 emotions, anger, fear, happy, sadness and surprise. The dataset is taken from https://github.com/WuJie1010/Facial-Expression. Recognition.Pytorch/tree/master/CK%2B48

The dataset consists of 750 files, 135 images for anger, 75 for fear, 207 for happy, 84 for sadness and 249 for surprise.

Figure 8 represents sample of pictures from the dataset.

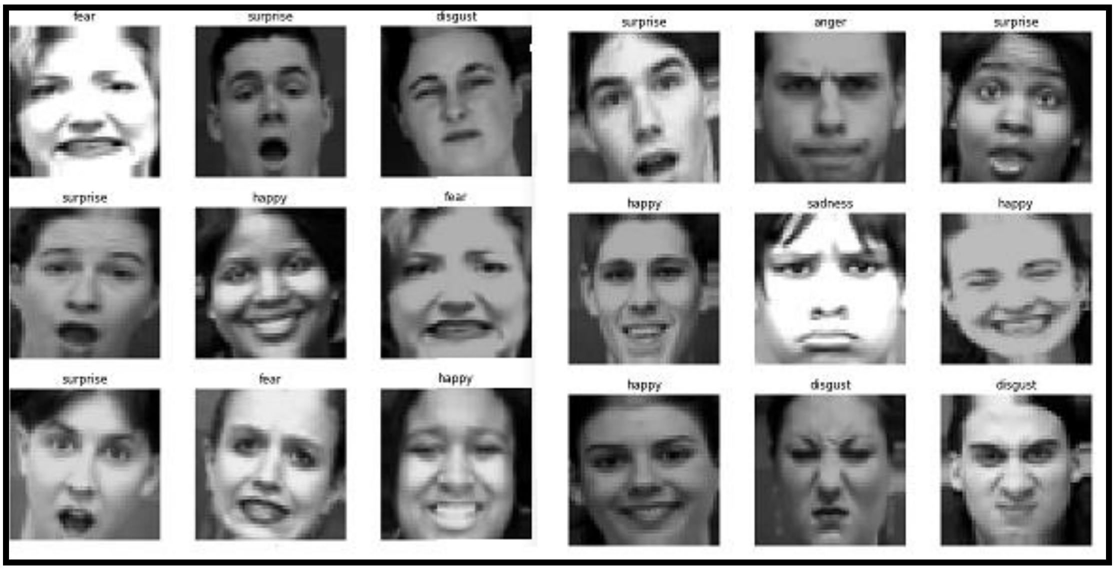

Figure 7 Sample of dataset pictures.

READING THE DATASET

```
INPUT_PATH = "CK+48/"

total_images = 0
for dir_ in os.listdir(INPUT_PATH):
    count = 0
    for f in os.listdir(INPUT_PATH + dir_ + "/"):
        count += 1
    total_images += count
    print(f"{dir_} has {count} number of images")

print(f"\ntotal images: {total_images}")
```

```
sadness has 84 number of images
happy has 207 number of images
anger has 135 number of images
surprise has 249 number of images
fear has 75 number of images

total images: 750
```

Figure 8 Reading the dataset.

DATA PREPARATION

Read every image in the dataset and prepare the labels for them

```
[7] TOP_EMOTIONS = ["happy", "surprise", "anger", "sadness", "fear"]

[8] img_arr = np.empty(shape=(total_images, 48, 48, 1))
    img_label = np.empty(shape=(total_images))
    label_to_text = {}

    idx = 0
    label = 0
    for dir_ in os.listdir(INPUT_PATH):
        if dir_ in  TOP_EMOTIONS:
            for f in os.listdir(INPUT_PATH + dir_ + "/"):
                img_arr[idx] = np.expand_dims(cv2.imread(INPUT_PATH + dir_ + "/" + f, 0), axis=2)
                img_label[idx] = label
                idx += 1
            label_to_text[label] = dir_
            label += 1

    img_label = np_utils.to_categorical(img_label)

    img_arr.shape, img_label.shape, label_to_text

((750, 48, 48, 1),
 (750, 5),
 {0: 'happy', 1: 'sadness', 2: 'anger', 3: 'surprise', 4: 'fear'})
```

Figure 9 Prepare data and labels.

Display the information of the dataset using the python:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 851264 entries, 0 to 851263
Data columns (total 7 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   date    851264 non-null  object
 1   symbol  851264 non-null  object
 2   open    851264 non-null  float64
 3   close   851264 non-null  float64
 4   low     851264 non-null  float64
 5   high    851264 non-null  float64
 6   volume  851264 non-null  float64
dtypes: float64(5), object(2)
memory usage: 45.5+ MB
```

Date & Symbol are in object datatype and rest are float datatype

Data set splitting into 70 % and 30 % for testing

```
[9] X_train, X_test, y_train, y_test = train_test_split(img_arr, img_label, train_size=0.70, stratify=img_label, shuffle=True,
    print(X_train.shape, X_test.shape)

(525, 48, 48, 1) (225, 48, 48, 1)
```

Figure 10 Dataset splitting.

Print some of data using the python

```
[10] idx = 0
     for k in label_to_text:
         sample_indices = np.random.choice(np.where(y_train[:,k]==1)[0], size=5, replace=False)
         sample_images = X_train[sample_indices]
         for img in sample_images:
             idx += 1
             ax = pyplot.subplot(5,5,idx)
             ax.imshow(img.reshape(48,48), cmap='gray')
             ax.set_xticks([])
             ax.set_yticks([])
             ax.set_title(label_to_text[k])
             pyplot.tight_layout()
```

Figure 11 Some of images in the dataset.

Data normalization by dividing each array of image on 255 to make the all values ranges from 0 to 1.

```
# data normalization
X_train = X_train / 255.
X_test = X_test / 255.
```

CNN model creation

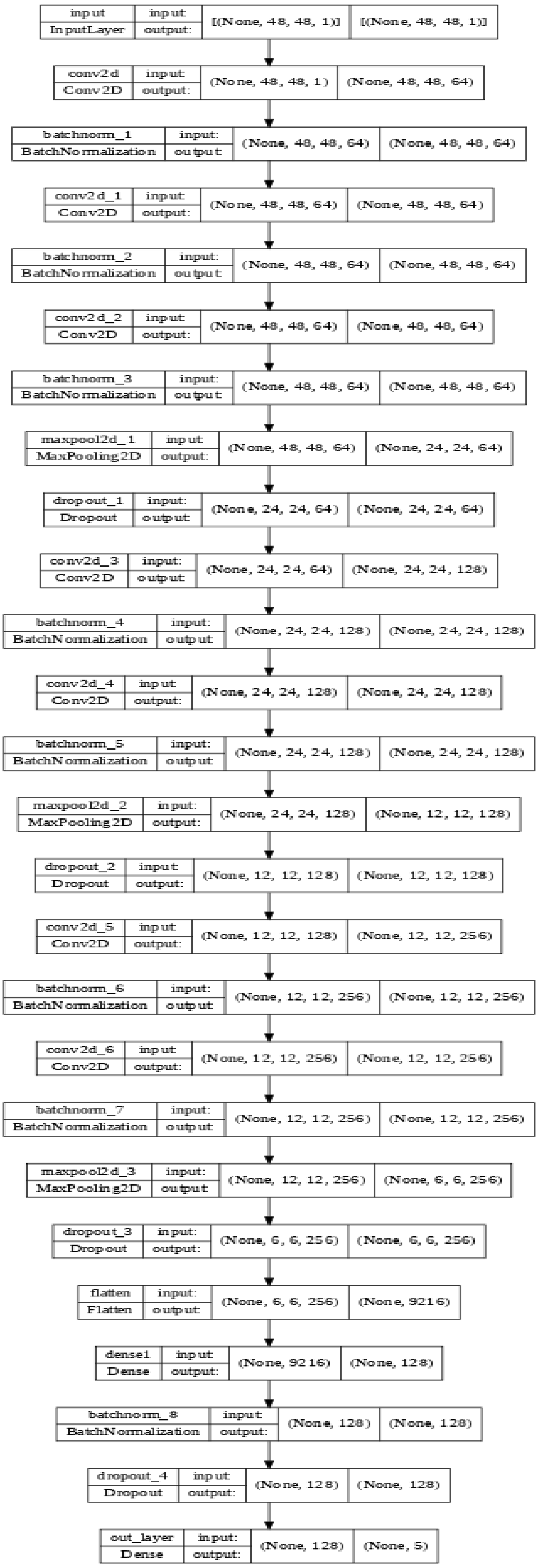
input | InputLayer | input: | output: | [(None, 48, 48, 1)] | [(None, 48, 48, 1)]
conv2d | Conv2D | input: | output: | (None, 48, 48, 1) | (None, 48, 48, 64)
batchnorm_1 | BatchNormalization | input: | output: | (None, 48, 48, 64) | (None, 48, 48, 64)
conv2d_1 | Conv2D | input: | output: | (None, 48, 48, 64) | (None, 48, 48, 64)
batchnorm_2 | BatchNormalization | input: | output: | (None, 48, 48, 64) | (None, 48, 48, 64)
conv2d_2 | Conv2D | input: | output: | (None, 48, 48, 64) | (None, 48, 48, 64)
batchnorm_3 | BatchNormalization | input: | output: | (None, 48, 48, 64) | (None, 48, 48, 64)
maxpool2d_1 | MaxPooling2D | input: | output: | (None, 48, 48, 64) | (None, 24, 24, 64)
dropout_1 | Dropout | input: | output: | (None, 24, 24, 64) | (None, 24, 24, 64)
conv2d_3 | Conv2D | input: | output: | (None, 24, 24, 64) | (None, 24, 24, 128)
batchnorm_4 | BatchNormalization | input: | output: | (None, 24, 24, 128) | (None, 24, 24, 128)
conv2d_4 | Conv2D | input: | output: | (None, 24, 24, 128) | (None, 24, 24, 128)
batchnorm_5 | BatchNormalization | input: | output: | (None, 24, 24, 128) | (None, 24, 24, 128)
maxpool2d_2 | MaxPooling2D | input: | output: | (None, 24, 24, 128) | (None, 12, 12, 128)
dropout_2 | Dropout | input: | output: | (None, 12, 12, 128) | (None, 12, 12, 128)
conv2d_5 | Conv2D | input: | output: | (None, 12, 12, 128) | (None, 12, 12, 256)
batchnorm_6 | BatchNormalization | input: | output: | (None, 12, 12, 256) | (None, 12, 12, 256)
conv2d_6 | Conv2D | input: | output: | (None, 12, 12, 256) | (None, 12, 12, 256)
batchnorm_7 | BatchNormalization | input: | output: | (None, 12, 12, 256) | (None, 12, 12, 256)
maxpool2d_3 | MaxPooling2D | input: | output: | (None, 12, 12, 256) | (None, 6, 6, 256)
dropout_3 | Dropout | input: | output: | (None, 6, 6, 256) | (None, 6, 6, 256)
flatten | Flatten | input: | output: | (None, 6, 6, 256) | (None, 9216)
dense1 | Dense | input: | output: | (None, 9216) | (None, 128)
batchnorm_8 | BatchNormalization | input: | output: | (None, 128) | (None, 128)
dropout_4 | Dropout | input: | output: | (None, 128) | (None, 128)
out_layer | Dense | input: | output: | (None, 128) | (None, 5)

Figure 12 CNN model

Training the model with last five epoch results

```
history = model.fit_generator(
    train_datagen.flow(X_train, y_train, batch_size=batch_size),
    validation_data=(X_test,y_test),
    steps_per_epoch=len(X_train) / batch_size,
    epochs=epochs,
    callbacks=callbacks,
    use_multiprocessing=True
)
Epoch 73/80
52/52 [==============================] - 2s 38ms/step - loss: 0.1014 - accuracy: 0.9752 - val_loss: 0.0298 - val_accuracy: 0.9822 - lr: 1.6384e-06
Epoch 74/80
52/52 [==============================] - 2s 39ms/step - loss: 0.1715 - accuracy: 0.9505 - val_loss: 0.0303 - val_accuracy: 0.9822 - lr: 1.6384e-06
Epoch 75/80
52/52 [==============================] - 2s 39ms/step - loss: 0.1186 - accuracy: 0.9581 - val_loss: 0.0308 - val_accuracy: 0.9822 - lr: 1.6384e-06
Epoch 76/80
53/52 [==============================] - ETA: 0s - loss: 0.1379 - accuracy: 0.9524Restoring model weights from the end of the best epoch: 64.
52/52 [==============================] - 2s 42ms/step - loss: 0.1379 - accuracy: 0.9524 - val_loss: 0.0309 - val_accuracy: 0.9822 - lr: 1.6384e-06
```

As shown in the above results the validation accuracy is equal to 98.22 % with loss =0.0309.

Training Vs Validation Accuracy and Loss

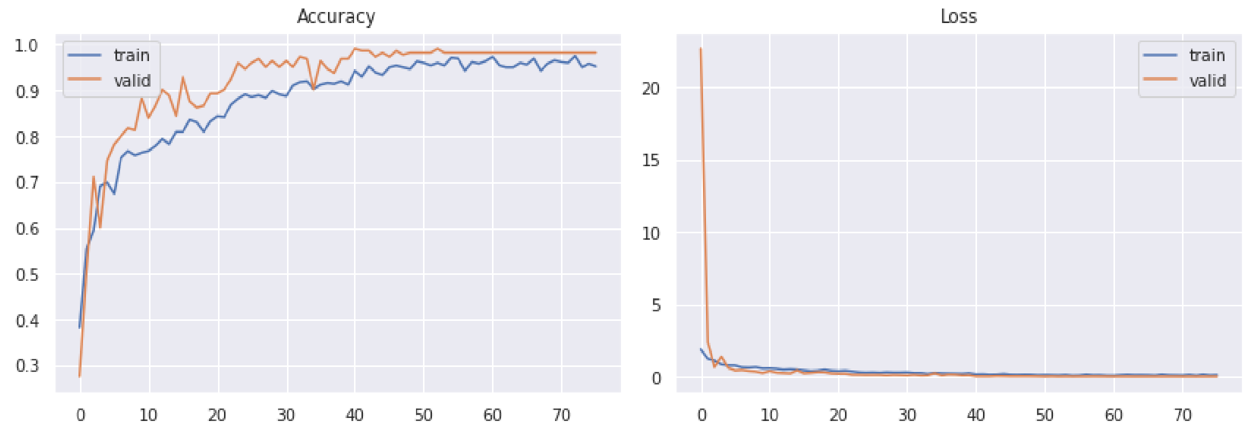

Figure 13 Training Vs. Validation Accuracy and loss.

Test accuracu and Classification report

test accuracy: 98.2222 %

	precision	recall	f1-score	support
0	1.00	1.00	1.00	62
1	0.92	0.92	0.92	25
2	0.95	0.95	0.95	40
3	1.00	1.00	1.00	75
4	1.00	1.00	1.00	23
accuracy			0.98	225
macro avg	0.97	0.97	0.97	225
weighted avg	0.98	0.98	0.98	225

Confusion Matrix

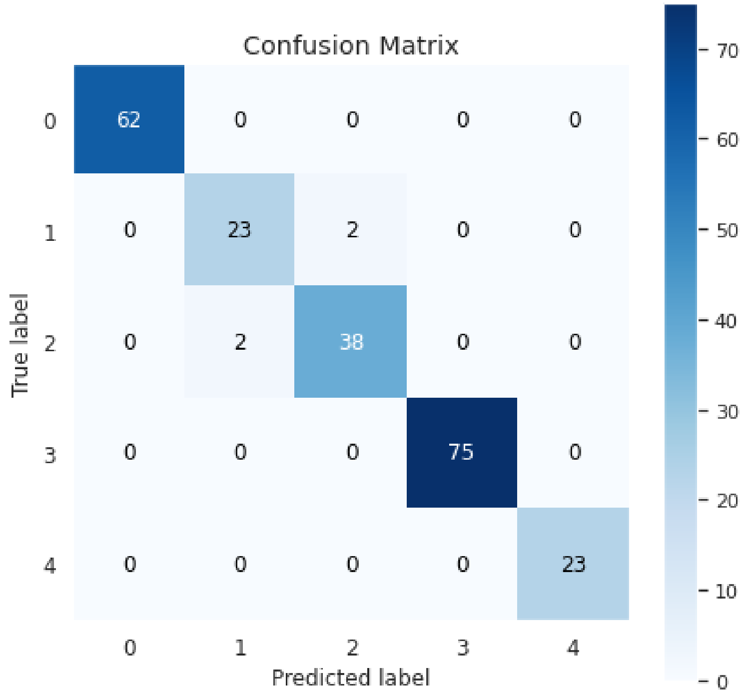

Figure 14 Confusion Matrix.

CONCLUSION

In this study, a convolutional neural network was created to recognize emotion in faces photographed in grayscale. Conv-ReLU-Pool architecture and softmax loss were employed. We experimented with batch normalization, L2 regularization, and dropout for regularization, but only the first two were kept in our final net. We employed ADAM with learning rate decay to optimize the network. Using this design, we were able to achieve a final validation accuracy of 98.22 percent. In the future work , we could implement the LSTM with other neural network to achieve accuracy of 100%.

REFERENCES

[1] Shervin Minaee, Mehdi Minaei, Amirali Abdolrashidi "Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network" University of California, Riverside [2019] .

[2] Wafa Mellouka, Wahida Handouzia "Facial emotion recognition using deep learning: review and insights" published by http://www.sciencedirect.com/, Leuven, Belgium, [2020].

[3] Ng HW, Nguyen VD, Vonikakis V, Winkler S Deep learning for emotion recognition on small datasets using transfer learning. In: Proceedings of the 2015 ACM on international (2015).

[4] conference on multimodal interaction. pp. 443–449.

[5] Xu M, Cheng W, Zhao Q, Ma L, Xu F Facial expression recognition based on transfer learning from deep convolutional networks. In: Proceedings of 11th international conference on natural computation, Zhangjiajie, China. pp 702–708, (2015).

[6] Fan Y, Lam JC, Li VO Multi-region ensemble convolutional neural network for facial expression recognition. In: Proceedings of International conference on artificial neural networks, Rhodes, Greece. pp 84–94, (2018).

[7] Wang Y, Li Y, Song Y, Rong X Facial expression recognition based on auxiliary models. Algorithms 12(11):227, (2019).

[8] Norde´n F, von Reis Marlevi F A comparative analysis of machine learning algorithms in binary facial expression recognition (Dissertation). http://www.divaportal.org/smash/record.jsf?pid=diva2%3A1329976&dswid=3676, (2019).

[9] Jyostna Devi B, Veeranjaneyulu N Facial emotion recognition using deep CNN based features. Int J Innov Technol Explor Eng (IJITEE), Vol. 8, No. 7, (2019).

[10] Nithya Roopa S Emotion recognition from facial expressions using deep learning. Int J Eng Adv Technol (IJEAT) 8(6S):91–65, (2019).

[11] Sreelakshmi P, Sumithra Facial expression recognition to robust to partial occlusion using MobileNet. Int J Eng Res Technol (IJERT) Vol. 8. No. 06, (2019).

[12] Ravi A Pre-trained convolutional neural network features for facial expression recognition. arXiv preprint arXiv:1812. 06387, (2018).

[13] Shaees, S, Naeem H, Arslan M, Naeem MR, Ali SH, Aldabbas H Facial emotion recognition using transfer learning. In: 2020 International conference on computing and information technology (ICCIT-1441). IEEE. pp. 1–5, (2020).

[14] Gulati N, Arun Kumar D Facial expression recognition with convolutional neural networks. Int J Future Gener Commun Netw 13(3):1923–1931, (2020).

[15] Ozdemir MA, Elagoz B, Alaybeyoglu A, Sadighzadeh R, Akan A Real time emotion recognition from facial expressions using CNN architecture. In: Proceedings of International Conference on medical technologies national congress, Kusadasi, Turkey. pp 1–4, (2019).

[16] Dhankhar P ResNet-50 and VGG-16 for recognizing Facial Emotions. Int J Innov Eng Technol 13(4):126–130, (2019).

[17] Picard RW Affective computing for HCI. In: HCI (1): 829–833,(1999).

[18] Daily SB, James MT, Cherry D, Porter III JJ, Darnell SS, Isaac J, Roy T Affective computing: historical foundations, current applications, and future trends. In: Emotions and affect in human factors and human-computer interaction, vol. 1, pp 213–231, (2017).

[19] Tao J, Tan T Affective computing: a review. In: International conference on affective computing and intelligent interaction. Springer, Berlin, Heidelberg, pp. 981–995, (2005).

[20] Thakare PP, Patil PS Facial expression recognition algorithm based on KNN classifier. Int J Comput Sci and Netw 5(6):941.

[21] Anirudha Ghosh, Abu Sufian, Farhana Sultana, Amlan Chakrabarti, Debashis De "Fundamental Concepts of Convolutional Neural Network", India, [2020].

[22] Mark Lutz "Learning Python" Fourth Edition, [2009] .